CLAIMS

What is claimed is:

1. A method of detecting a malware comprising the steps of:

interrupting execution of a process that has been loaded for execution;

scanning the process for a malware;

allowing the process to execute, if no malware is found; and

terminating execution of the process, if a malware is found.

2. The method of claim 1, wherein the process is associated with an application program.

3. The method of claim 1, wherein the process is loaded from at least one compressed, packed, or encrypted file.

4. The method of claim 1, wherein execution of the process comprises the step of:

loading code for execution by the process from at least one compressed, packed, or encrypted file.

5. The method of claim 4, wherein the step of interrupting execution of the process comprises the step of:

interrupting execution of the process when the process accesses at least one file that is not needed to perform decryption, decompression, or unpacking.

6. The method of claim 5, wherein the at least one file that is not needed to perform decryption, decompression, or unpacking comprises a system library file.

7. The method of claim 5, wherein the at least one file that is not needed to perform decryption, decompression, or unpacking comprises an executable file not related to the process.

8. The method of claim 5, wherein the at least one file that is not needed to perform decryption, decompression, or unpacking comprises a data file not related to the process.

9. The method of claim 5, wherein the malware is a computer virus.

10. The method of claim 5, wherein the malware is a computer worm.

11. The method of claim 5, wherein the malware is a Trojan horse program.

12. The method of claim 5, further comprising the step of:

scanning the process for a malware before execution of the process.

13. A system for detecting a malware comprising:

a processor operable to execute computer program instructions;

a memory operable to store computer program instructions executable by the processor; and

computer program instructions stored in the memory and executable to perform the steps of:

interrupting execution of a process that has been loaded for execution;

scanning the process for a malware;

allowing the process to execute, if no malware is found; and

terminating execution of the process, if a malware is found.

14. The system of claim 13, wherein the process is associated with an application program.

15. The system of claim 13, wherein the process is loaded from at least one compressed, packed, or encrypted file.

16. The system of claim 13, wherein execution of the process comprises the step of:

loading code for execution by the process from at least one compressed, packed, or encrypted file.

17. The system of claim 16, wherein the step of interrupting execution of the process comprises the step of:

interrupting execution of the process when the process accesses at least one file that is not needed to perform decryption, decompression, or unpacking.

18. The system of claim 17, wherein the at least one file that is not needed to perform decryption, decompression, or unpacking comprises a system library file.

19. The system of claim 17, wherein the at least one file that is not needed to perform decryption, decompression, or unpacking comprises an executable file not related to the process.

20. The system of claim 17, wherein the at least one file that is not needed to perform decryption, decompression, or unpacking comprises a data file not related to the process.

21. The system of claim 17, wherein the malware is a computer virus.

22. The system of claim 17, wherein the malware is a computer worm.

23. The system of claim 17, wherein the malware is a Trojan horse program.

24. The system of claim 17, further comprising the step of:

scanning the process for a malware before execution of the process.

25. A computer program product for detecting a malware comprising:

a computer readable medium;

computer program instructions, recorded on the computer readable medium, executable by a processor, for performing the steps of

interrupting execution of a process that has been loaded for execution;

scanning the process for a malware;

allowing the process to execute, if no malware is found; and

terminating execution of the process, if a malware is found.

26. The computer program product of claim 25, wherein the process is associated with an application program.

27. The computer program product of claim 25, wherein the process is loaded from at least one compressed, packed, or encrypted file.

28. The computer program product of claim 25, wherein execution of the process comprises the step of:

loading code for execution by the process from at least one compressed, packed, or encrypted file.

29. The computer program product of claim 28, wherein the step of interrupting execution of the process comprises the step of:

interrupting execution of the process when the process accesses at least one file that is not needed to perform decryption, decompression, or unpacking.

30. The computer program product of claim 29, wherein the at least one file that is not needed to perform decryption, decompression, or unpacking comprises a system library file.

31. The computer program product of claim 29, wherein the at least one file that is not needed to perform decryption, decompression, or unpacking comprises an executable file not related to the process.

32. The computer program product of claim 29, wherein the at least one file that is not needed to perform decryption, decompression, or unpacking comprises a data file not related to the process.

33. The computer program product of claim 29, wherein the malware is a computer virus.

34. The computer program product of claim 29, wherein the malware is a computer worm.

35. The computer program product of claim 29, wherein the malware is a Trojan horse program.

36. The computer program product of claim 29, further comprising the step of:

scanning the process for a malware before execution of the process.